Test of Pearl 000 — Binary logic and computer architecture

Pearls of Computer Science (201700139)
Bachelor module 1.1, Technical Computer Science, EWI
September 6, 2019, 13:45–14:45

Module coordinator: Doina Bucur
Instructor: Pieter-Tjerk de Boer

- You may use 1 A4 document with your own notes for this exam and a *simple* calculator.
- Scientific or graphical calculators, laptops, mobile phones, books etc. are not allowed. ***Put those in your bag now!***
- Questions marked with (MC) must be answered on the separate multiple-choice form, at the number indicated in the circle.
- Other questions have a box in which you can write the answer on this paper; this paper must be handed in.
- Total number of points: 100.
  Total number of pages: 6.

Your name:

(please underline your family name (i.e., the name on your student card), so that we know how to sort)

Your student number:

---

1. Binary numbers

4 pt (a) What is the decimal number –4 expressed as a 6-bit 2-complement binary number?

(MC**01**)

A. 100011
B. 100100
C. 100101
D. 111010
E. 111011
F. 111100
G. 111101

4 pt (b) Considering the 1-complement and 2-complement number representation scheme(s), which has/have the property that inverting all bits is the same as changing the sign?

(MC**02**)

A. Neither of them.
B. Only the 1-complement scheme.
C. Only the 2-complement scheme.
D. Both of them.

4 pt (c) Convert hexadecimal A0F to decimal.

(MC**03**)

A. 1015
B. 1515
C. 2264
D. 2265
E. 2491
F. 2575
G. 2830

4 pt (d) Convert –1 decimal to two-complements 2-digit hexadecimal (i.e., first convert to 2-complements 8-bit binary and then to 2-digit hexadecimal)

MC**04**

A. 00
B. 01
C. 1F
D. 11
E. F1
F. FE
G. FF

4 pt (e) Which of the following operations increases a binary number by 25%?
(You may assume the binary number is a multiple of 4, so 25% of it is still an integer.)

MC**05**

A. Shift to the left by 1 position.
B. Shift to the right by 2 positions.
C. Shift to the right by 25 positions.
D. Shift to the right by 2 positions and then to the left by 1 position.
E. Shift to the left by 2 positions and then add the original (unshifted) number to it.
F. Shift to the left by 2 positions and then subtract the original (unshifted) number from it.
G. Shift to the right by 2 positions and then add the original (unshifted) number to it.

4 pt (f) Suppose we have 4-bit adder for unsigned numbers, which has 4 output bits and a "carry" output; the latter is essentially the 5th output bit needed if the sum exceeds 15.
Now assume we want to use this adder to perform additions of 4-bit **1**-complement signed numbers. What should we do with the "carry" output to get correct results for both positive and negative numbers?
You may assume that the 1-complement input numbers are such that their sum is in the range –7 to +7, so it can be represented by the adder's 4 normal output bits.

MC**06**

A. Just ignore the carry bit.
B. If the carry bit is a 1, invert the 4 normal output bits.
C. If the carry bit is a 1, invert the most-significant of the 4 normal output bits.
D. If the carry bit is a 1, invert the least-significant of the 4 normal output bits.
E. If the carry bit is a 1, add 1 to the result represented by the 4 normal output bits.
F. If the carry bit is a 1, subtract 1 from the result represented by the 4 normal output bits.

---

2. Boolean logic

4 pt (a) Give the truth table of a two-input greater-than comparator: only if input X is greater than input Y, the output is 1. Note that you have to answer 4 multiple choice questions here; choose A for 0 or B for 1.

X	Y	output
0	0	MC**07**
0	1	MC**08**
1	0	MC**09**
1	1	MC**10**

8 pt (b) Consider the following derivation in Boolean algebra. For each step, indicate on the multiple-choice form which rule is applied, using the following options:

- A commutative
- B identity
- C complement
- D distributive
- E DeMorgan
- F this step is not correct

$$\overline{X+Y}+(\overline{X}\cdot(Y\cdot Z))$$

MC**11** $= (\overline{X}\cdot\overline{Y})+(\overline{X}\cdot(Y\cdot Z))$

MC**12** $= \overline{X}\cdot(\overline{Y}+(Y\cdot Z))$

MC**13** $= \overline{X}\cdot(\overline{Y}+Y)\cdot(\overline{Y}+Z)$

MC**14** $= \overline{X}\cdot 1\cdot\overline{Y+Z}$

MC**15** $= \overline{X}\cdot\overline{Y+Z}$

MC**16** $= \overline{X+Y}+Z$

6 pt (c) Sketch a diagram implementing the following formula with only NAND gates: $\overline{(\overline{A}\cdot B)+C}$

4 pt (d) Suppose you take two 2-input NAND gates, and feed their outputs into a third 2-input NAND gate. Does this as a whole work as a 4-input NAND gate?

MC**17**

A. Yes, it does.
B. No, a 4-input NAND gate is not well-defined.
C. No, for that we should replace the third NAND gate by an OR gate.
D. No, for that we should replace the third NAND gate by a NOR gate.
E. No, for that we should put inverters at the inputs of the first two NAND gates.
F. No, for that we should put inverters at the outputs of the first two NAND gates.

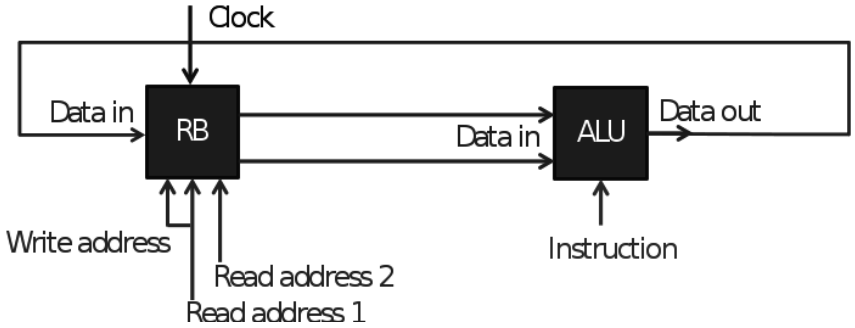
4 pt (e) Suppose you have a lot of 2-input NOR gates. Can you make an AND gate out of this?

MC**18**

A. Yes, thanks to DeMorgan's theorem.
B. Yes, thanks to the distributive rule.
C. Yes, thanks to the complementary rule.
D. No, you'd also need inverters.
E. No, you'd also need at least one OR gate.
F. No, you'd also need inverters and at least one OR gate.

---

12 pt **3. Problem 3**



The ALU of the processor above has two instructions: 0 = 'ADD' and 1 = 'MUL'. Furthermore it has four 8-bit registers. Give for this processor the program for computing R1 $\times$ R2 + R1 $\times$ R3 + R1 and storing the result into R1. (You may not need all timeslots.)

	read address 1 / write address	read address 2	instruction
Timeslot 0			
Timeslot 1			
Timeslot 2			
Timeslot 3			
Timeslot 4			
Timeslot 5			

Continued on next page...

4. Problem 4

Given this AVR program; “BRNE” means “BRanch if Not Equal”, “BREQ” means “BRanch if EQual”, “SUB” means “Subtract”.

```
LDI   R17, $04
LDI   R18, $08
LDI   R19, $06
LDI   R20, $08
SUB   R19, R18
BREQ  +1
MOV   R19, R20
SUB   R20, R17
BRNE  -5
```

Assume that each instruction takes 1 clock cycle, except jumping to a different address, which takes 2 clock cycles.

18 pt (a) Fill in the below table with the status of the registers after each instruction; if a register doesn’t change from one line to the next, you may leave it blank.

R17	R18	R19	R20	

5 pt (b) How many clockcycles does the program (of the previous page) take?
Explain.

15 pt **5. Problem 5**

What is the mathematical function that is computed by the code below?
Write as a function of X and Y, e.g. f(X, Y) = X + Y, and explain.
Assume that X and Y are larger than 0 and that X $<$ Y; the result is available in R20.

```
   LDI   R17, $X
   LDI   R18, $Y
   LDI   R19, $00
   LDI   R21, $01
label1:
   ADD   R19, R17
   ADD   R17, $01
   MOV   R20, R17
   SUB   R20, R18
   BRNE label1
   MOV   R20, R19
```

End of this exam.